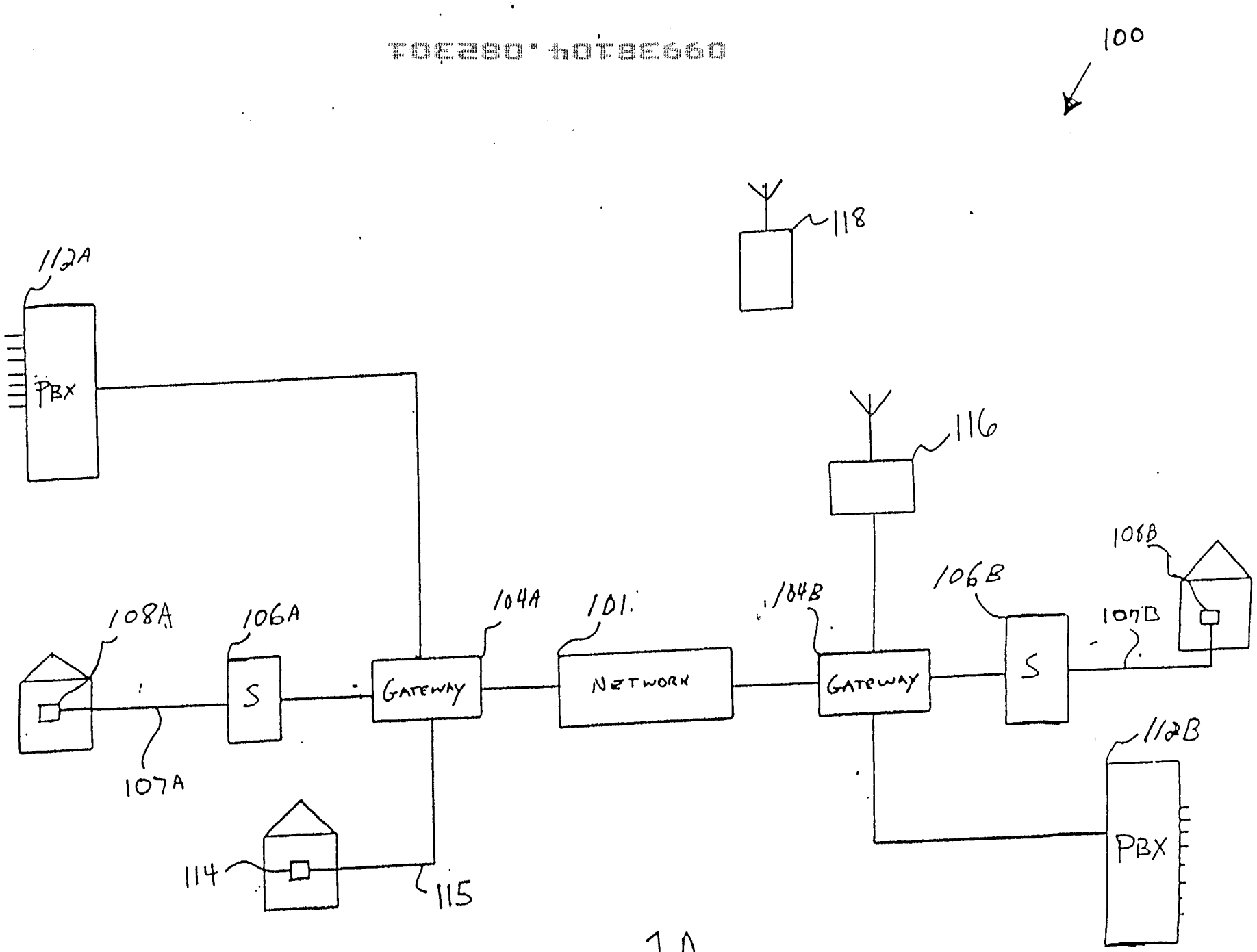


FIG. 1A

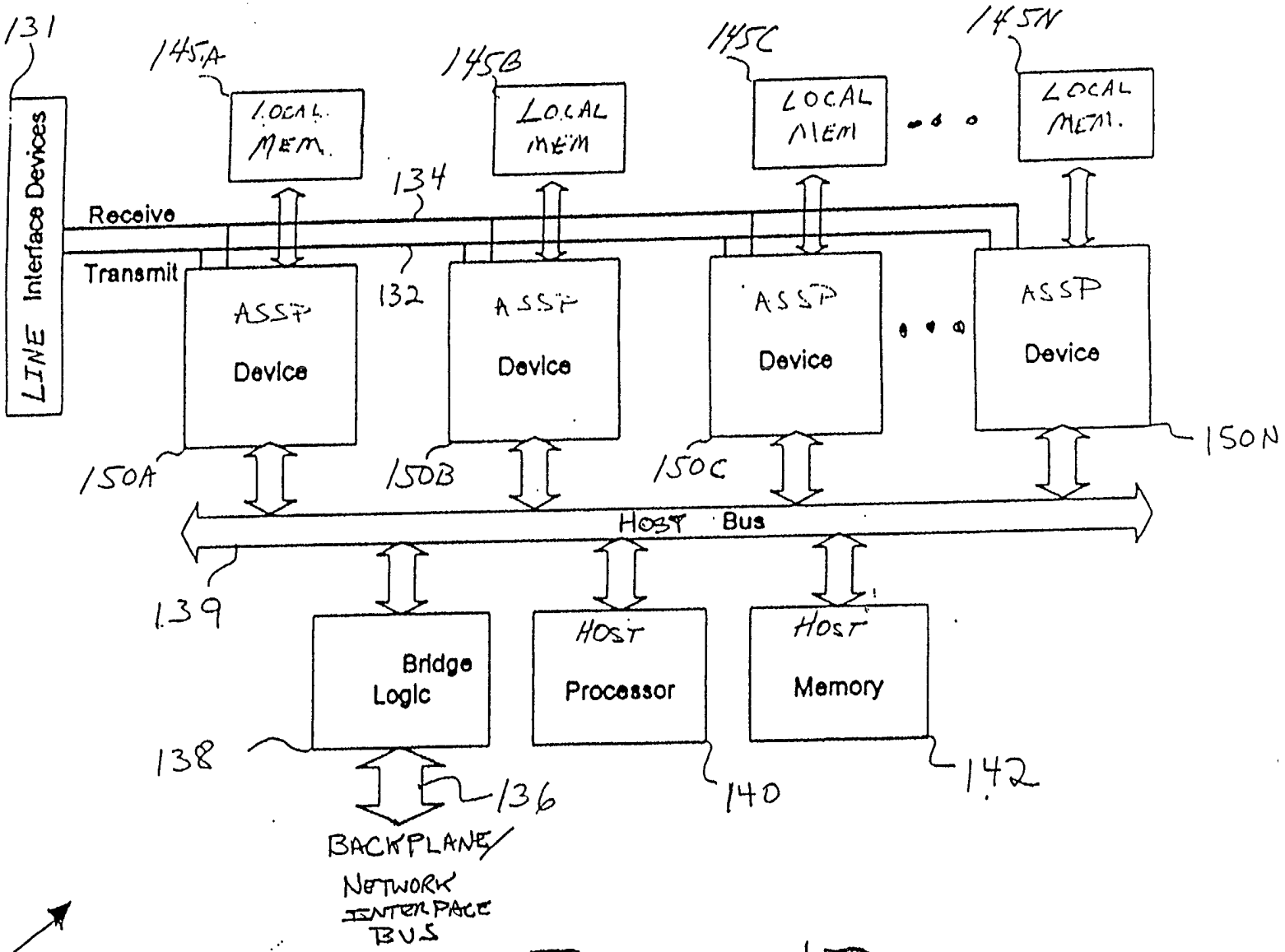


FIG. 1B

150

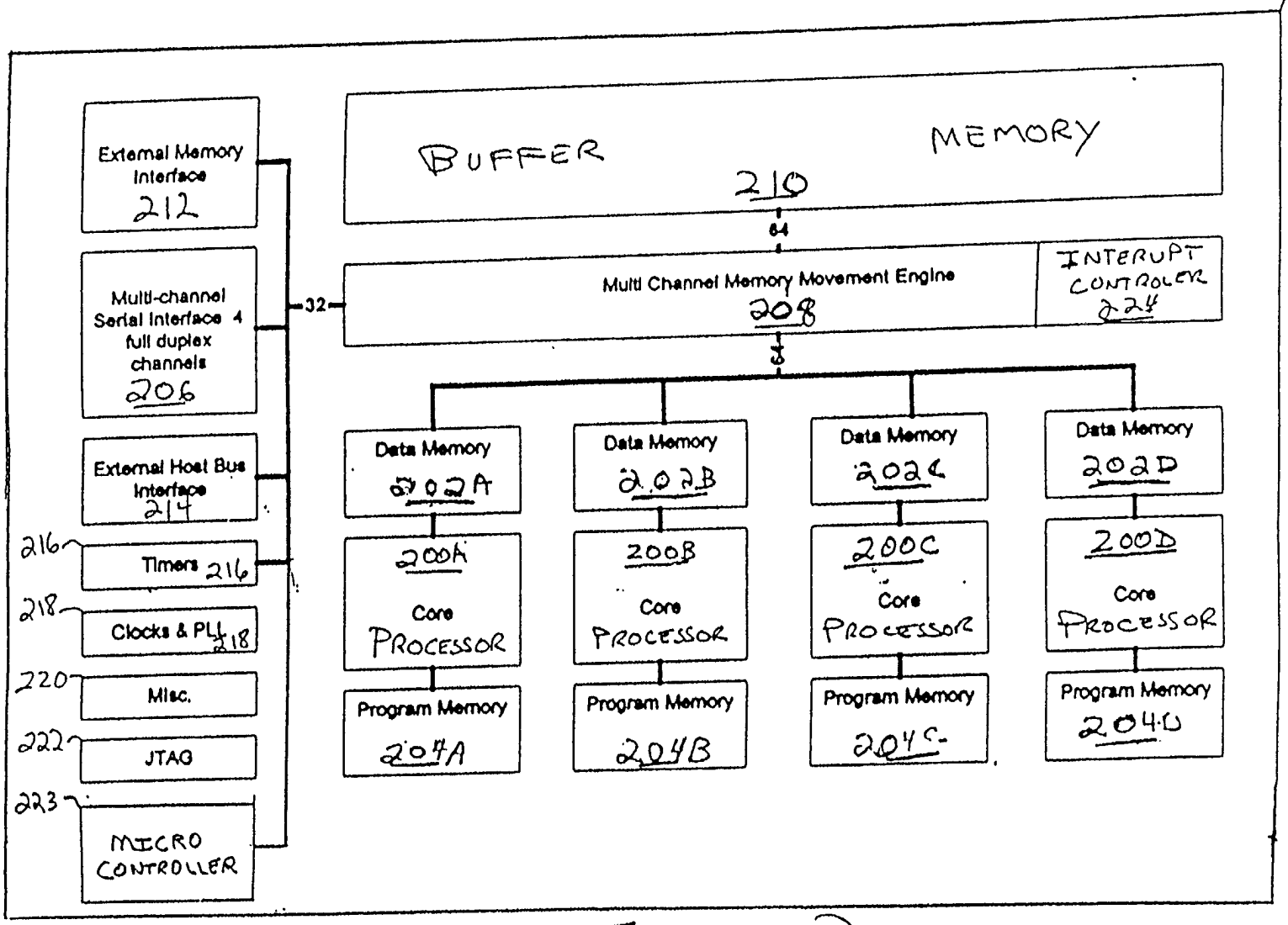


FIG. 2

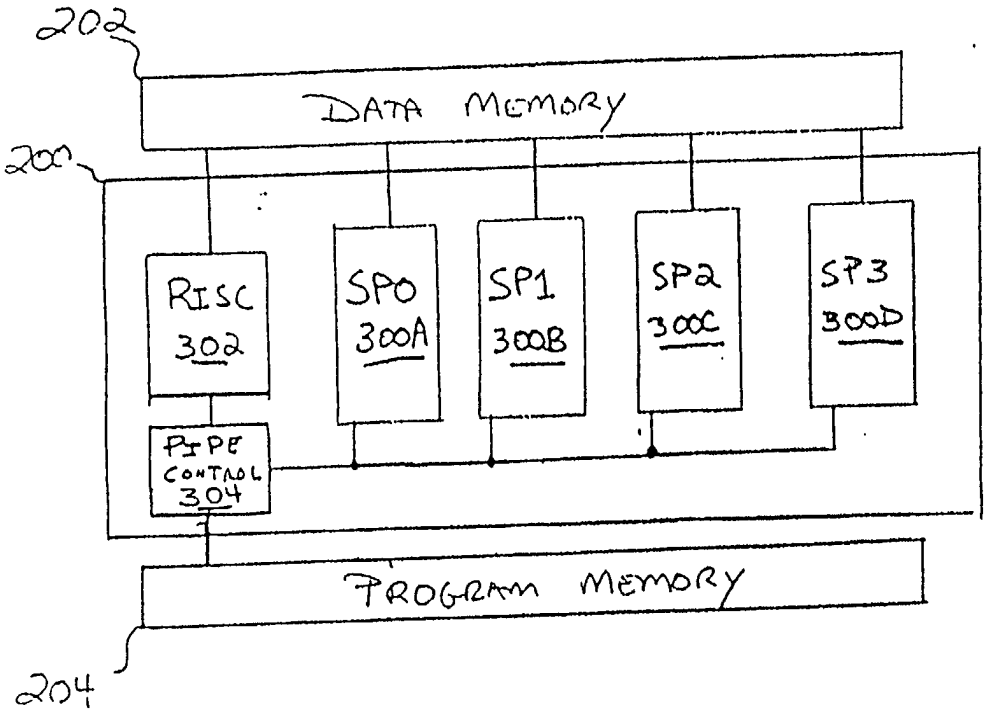


FIG. 3

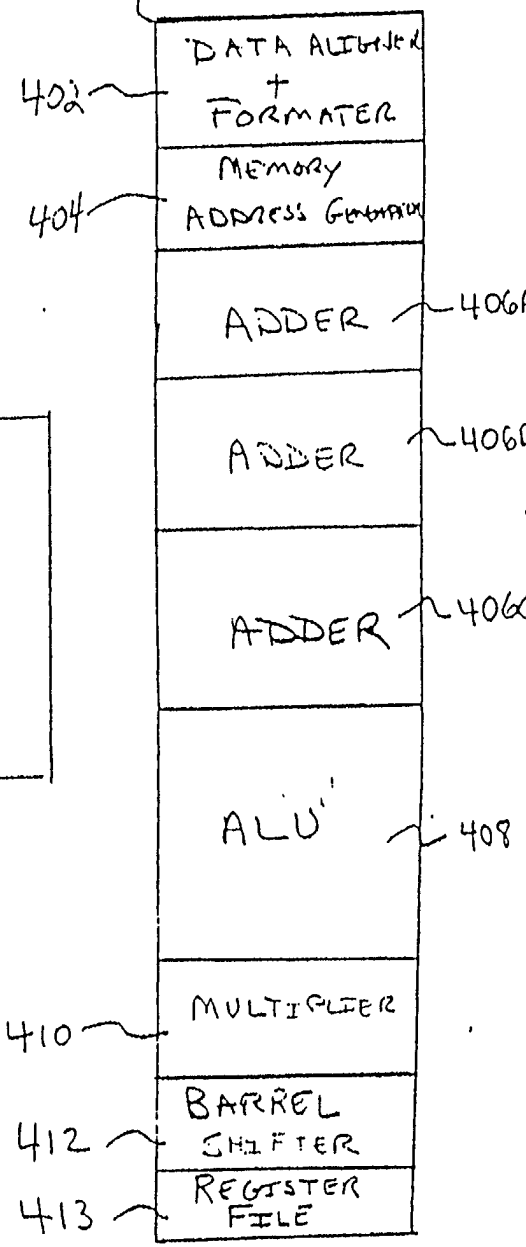


FIG. 4

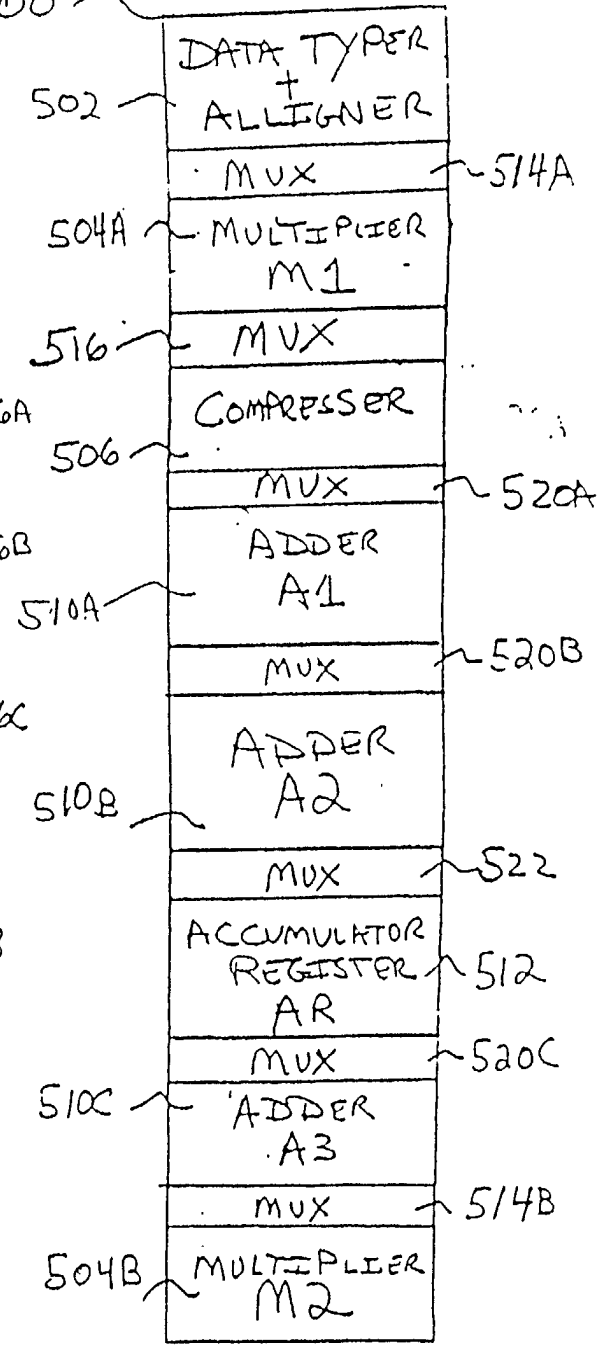


FIG. 5A

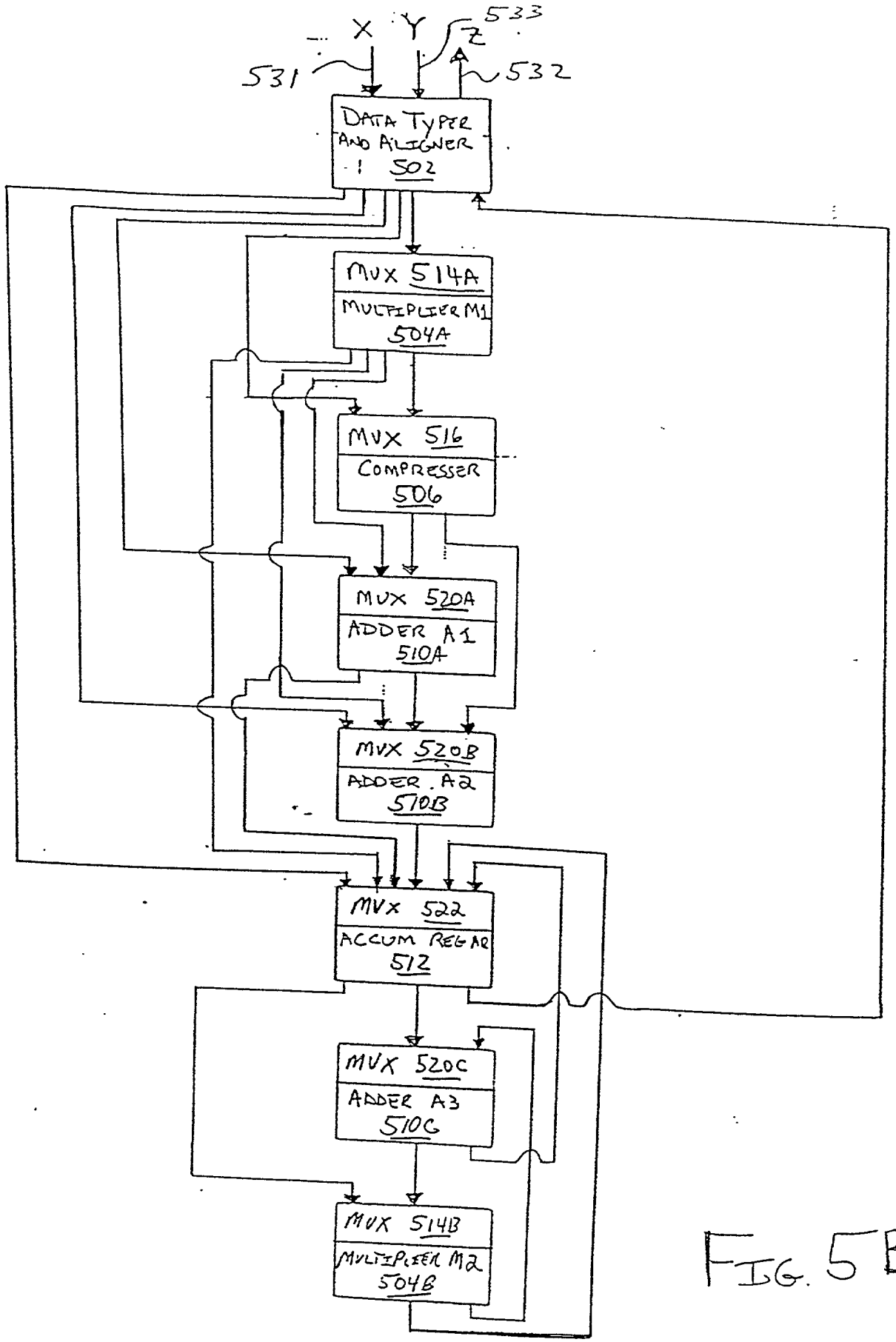


FIG. 5B

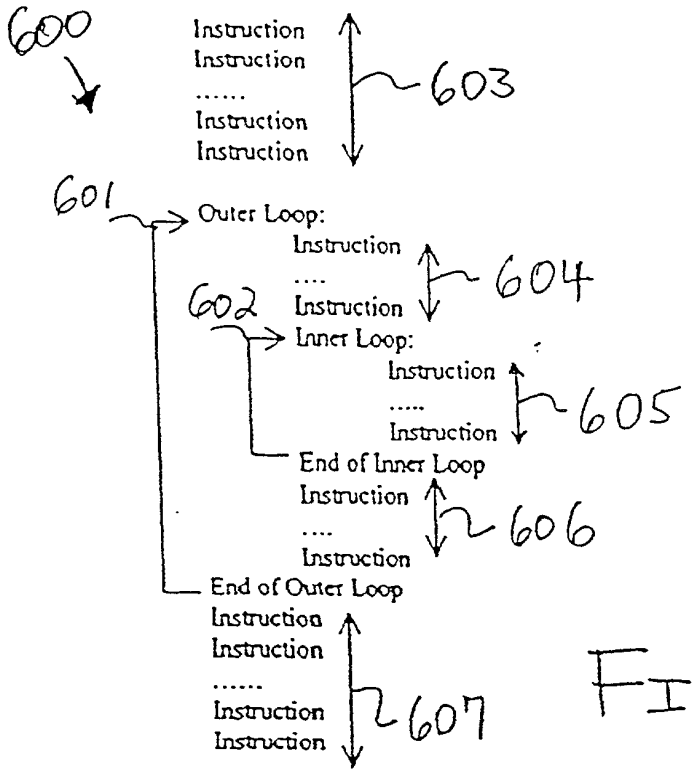


FIG. 6A

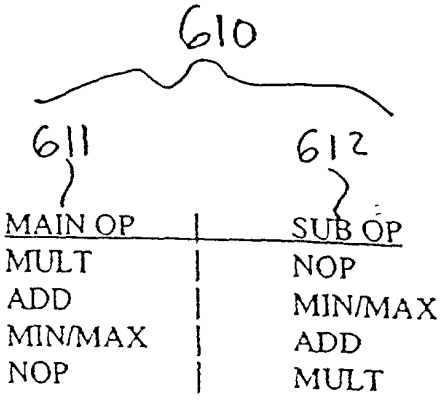


FIG. 6B

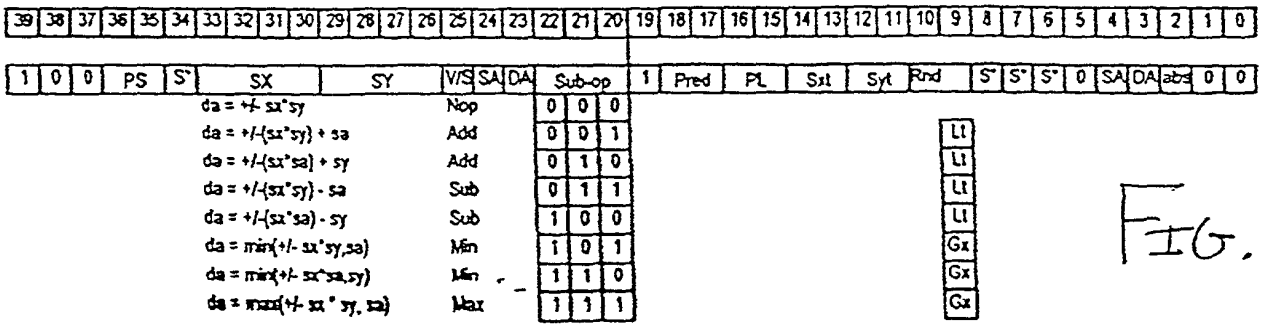


FIG. 6C

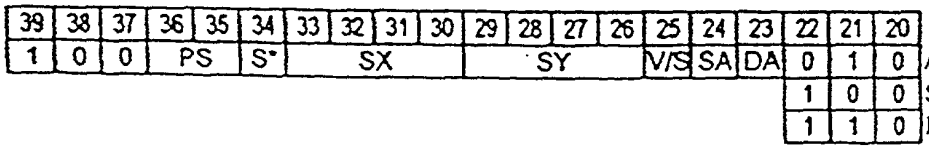


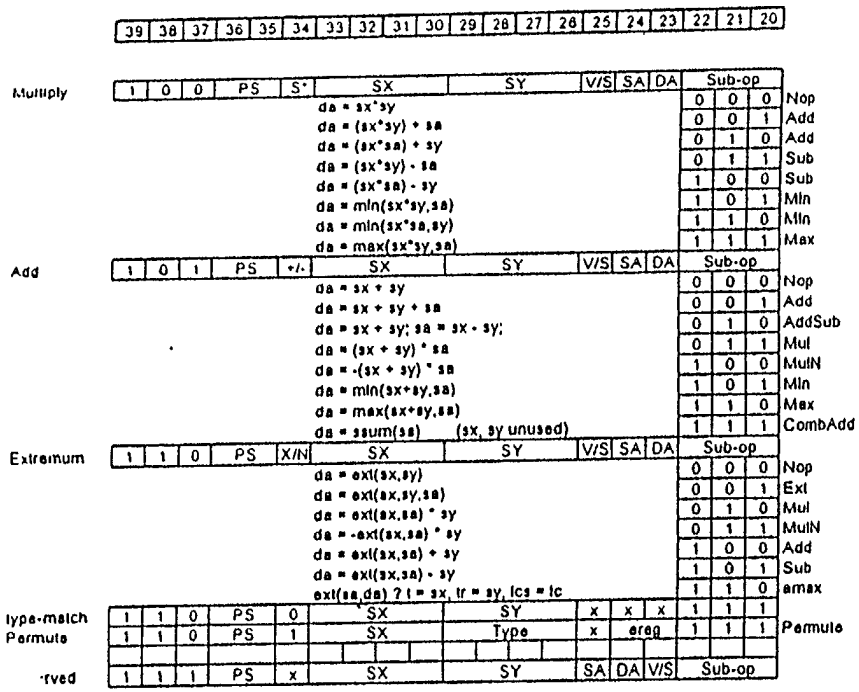
FIG. 6D

20-bit ISA

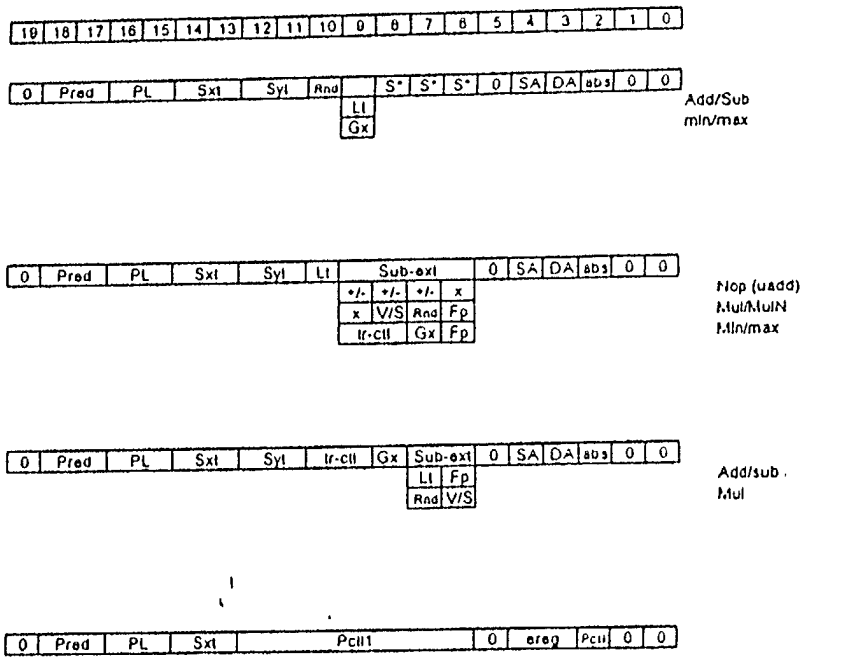
39	19	
0	0	20-bit parallel
0	1	20-bit serial
1	0	40-bit extended
1	1	20-bit serial

Control || Control
Control # Control
DSP, extensions/Shadow
DSP # DSP

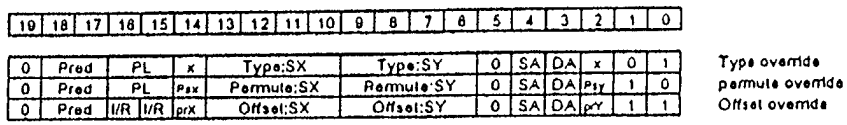
DSP Instructions



Control and specifier Extensions



Type/offset/permute extensions



Shadow DSP

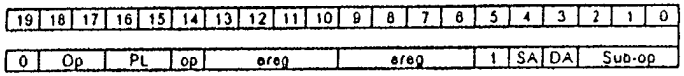


FIG. 6 E

Control Instructions

Instruction	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
add,sub	L	Pred		0	0	0	RX				RY				RZ				+/-	0
max,min	L	Pred		0	0	0	RX				RY				RZ				X/N	1
Shift	L	Pred		0	0	1	RX				UI4				RZ				UI1	R/L
Logic	L	Pred		0	1	0	RX				RY				RZ				&, |	&|
Mux	L	Pred		0	1	1	RX				RY				RZ				Pd	0
mov	L	Pred		0	1	1	RX				DZ				Rxt	Dzt	0	0	0	1
addi	L	Pred		0	1	1	SI4				DZ				x	x	1	0	0	1
mov2erg	L	Pred		0	1	1	RX				unit		ereg		gd	type		1	0	1
Ldm	L	Pred		0	1	1	RX				DZ1				DZ2				1	1
4bits	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI4				0
bits	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI2		0	0	1
Setbit	L	Pred		1	0	0	UI4:POS				RZ				Rzt	UI1	UI1	1	0	1
Movi	L	Pred		1	0	0	SI8								RZ				-1	1
Jmp	L	Pred		1	0	1	SI9									0	Pred		0	0
Call	L	Pred		1	0	1	SI9									1	Pred		0	0
Loop	L	Pred		1	0	1	UI5 : Lcount					UI5: Lsize					UI2: Lsl		0	1
Jmpi	L	Pred		1	0	1	RX				x	x	x	x	x	0	Pred		1	0
Calli	L	Pred		1	0	1	RX				x	x	x	x	x	1	Pred		1	0
Loopi	L	Pred		1	0	1	RX				x	UI5: Lsize					UI2: Lsl		1	1
Test	L	Pred		1	1	0	RX				RY				PZ			=, <, >		0
Testbit	L	Pred		1	1	0	RX				UI5					PZ		B	0	1
Andp, orp	L	Pred		1	1	0	Pa			Pb			Pc		PZ			&|	1	1
Load	L	Pred		1	1	1	MX				RZ				Ext			0	0	0
Store	L	Pred		1	1	1	MZ				RX				Ext			1	0	0
eLoad	L	Pred		1	1	1	MX				RZ				1	1	1	0	0	0
eStore	L	Pred		1	1	1	MZ				RX				1	1	1	1	0	0
Extended	L	Pred		1	1	1	Bits 27:18												1	0
Logic2	L	Pred		1	1	1	RX				RY/RZ				Rxt	Ryt	&, |, &|, |		0	1
mov-erg	L	Pred		1	1	1	unit		ereg		RZ				gd	Sft		0	1	1
Crb	L	Pred		1	1	1	RX				RZ				s/m	0	0	1	1	1
Parity	L	Pred		1	1	1	RX				PZ			O/E	0	1	0	1	1	1
Sim	L	Pred		1	1	1	MZ				RX				1	1	0	1	1	1
Abs	L	Pred		1	1	1	'RX				RZ				0	0	1	1	1	1
Neg	L	Pred		1	1	1	RX				RZ				0	1	1	1	1	1
Div-step	L	Pred		1	1	1	RX				RZ				1	0	1	1	1	1
& Set	L	Pred		1	1	1	RX				PZ			0	1	1	1	1	1	1
	[illegible]																			
Return	L	Pred		1	1	1	Pred		I-ctl		0	1	0	1	1	1	1	1	1	1
Zero-ac	L	Pred		1	1	1	ac #				1	1	0	1	1	1	1	1	1	1
eSync	L	Pred		1	1	1	RZ				0	1	1	1	1	1	1	1	1	1
Swi	L	Pred		1	1	1	UI3			0	1	1	1	1	1	1	1	1	1	1
Nop	L	Pred		1	1	1	UI3			1	1	1	1	1	1	1	1	1	1	1

<Bit1, Bits9-6> == UI5 (Shift Amount)

<Bit3, Bits13-10> == UI5 :POS

FIG. 6F

Extended Control

	Bits 13.2 of upper half (39.20)																															
	13	12	11	10	9	8	7	6	5	4	3	2	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Insert/Extract	RX				RZ				0	0	0	0	0	x	x	0	Rxt	Rzt	I/E	R/I	R/I	Offset:UI5					Length:UI5					0
																						x	RY				RV				x	
Inserti	UI4: length				RZ				0	0	0	1	0	x	x	0	Rzt	UI5: Position					Imm10									
Shift	RX				RZ				0	0	0	0	0	rxh	rxl	0	rzh	rzl	D	U/S	1	Shift: UI5					A/L	LI	R/L	0	Fill	1
																					0	ryh	RY									
Rotate	RX				RZ				0	0	0	0	0	x	x'	0	x	x	x	x	1	Shift: UI5					1	x	R/L	1	x	1
																					0	ryh	RY									
jmp, call	UI7							J/C	0	0	1	0	0	Pred		0	x	UI15														
dloop	UI4: outer LC				UI4: Inner LC				0	0	1	1	0	x	exit	0	UI1	UI4: outer Lsize				UI4: inner Lsize				UI2: O-Ls		UI4: inner Lstart				0
dloopi	RX				RY				0	0	1	1	0	x	exit	0	x	UI4: outer Lsize				UI4: inner Lsize				UI2: O-Ls		UI4: inner Lstart				1
4	RX				RY				0	1	0	0	0	x	x	0	0	rxh	rnd	ryh	+/-	a/+	RZ				I/f	rzh	rzl	s/u	s/u	0
sub	RX				RY				0	1	0	0	0	x	x	0	0	rxh	rxl	ryh	ryl	+/-	RZ				LI	rzh	rzl	x	x	1
reserved	[illegible]																															
logicp	PX				D	PZ			0	1	0	0	0	x	x	0	1	T/F	T/F	T/F	&|	&|	PY				PV				x	1
Testi	RX				D	PZ			0	1	0	1	0	=,>,<		0	Imm16															
Movi	H/L	Fill			RZ				0	1	1	0	0	x	x	0	Imm16															
loadi	Type				RZ				0	1	1	1	0	x	x	0	0	0	Imm14													
storei	Type				RX				0	1	1	1	0	x	x	0	0	1	Imm14													
loadi	RX				RZ				0	1	1	1	0	x	x	0	1	Rzt	0	Type			SI10									
storei	MZ				RX				0	1	1	1	0	x	x	0	1	Rzt	1	Type			SI10									
Add/subi	RX				RZ				1	0	+/-	0	0	LI	s/u	0	Imm16															
mini,maxi	RX				RZ				1	0	X/N	1	0	x	x	0	Imm18															
andi,ori	RX				RZ				1	1	&|	H/L	0	x	x	0	Imm16															

Fill: Sign/Zero

Bit 15 is continuation of Inner LC

andp, orp, andorp, orandp: pz = (px relop py) relop pv)

FIG. 6G

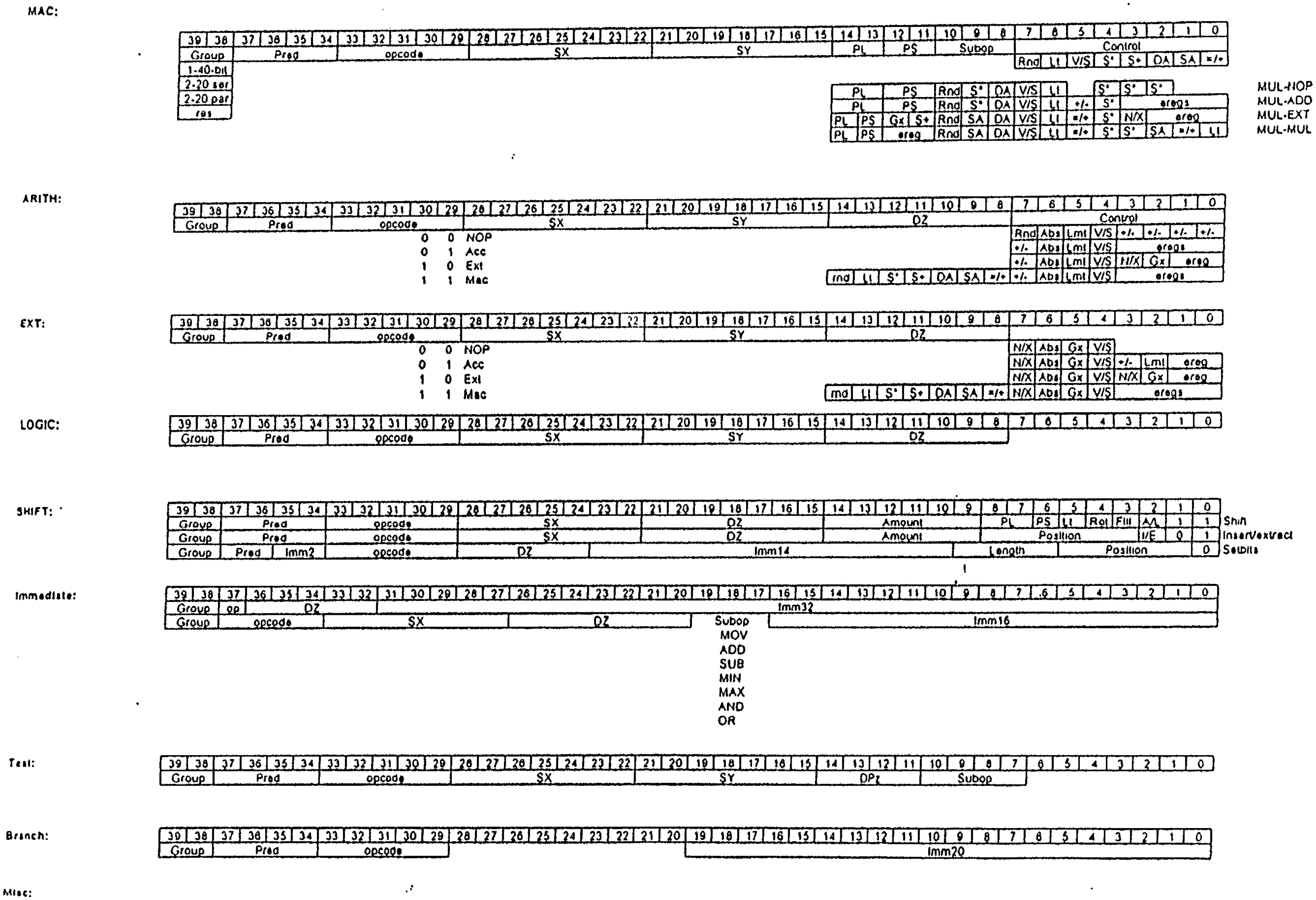


FIG. 6H

7-bit specifier: Parallel Store, Parallel Load in DSP Instructions

6	5	4	3	2	1	0
M/R						
0	0	0	SPR: s0-s15			
0	0	1	reserved			
0	1	0	as-names			
0	1	1	gpr, r0-r15			
1	0	ptr (r0) to (r15)				off
1	1	offset UI4				pV

Mem[ptr] || ptr += Idx — Always postupdate
Mem[ptr + Idx] — pV: p14, p15 — Always preupdate

6-bit specifier: DSP Instructions

5	4	3	2	1	0
M/R					
0	0	as-names			
0	1	gpr, r0-r15			
1	ptr (r0) to (r15)				off

Always postupdate

5-bit specifier: RISC Instructions

4	3	2	1	0
0	spr: s0-s15			
1	gpr: r0-r15			

4-bit specifier:

3	2	1	0	
gpr: r0-r15				RISC Instructions
pV: (r0-r7)			off	20-bit DSP Instructions
areg				20-bit Shadow DSP Instructions

AR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
type				permute					CB	Idx1, UI3 (0-7)			Idx0, SI6 (-16 to 15)					ptr													

as-names:

3	2	1	0	
0	0	0	0	A0 (use type, SIMD)
0	0	0	1	A1
0	0	1	0	T
0	0	1	1	TR
0	1	0	0	A00 (unit 0)
0	1	0	1	A10
0	1	1	0	T0
0	1	1	1	TR0
1	0	0	0	SX1
1	0	0	1	SX1a
1	0	1	0	SX2
1	0	1	1	SX2a
1	1	0	0	SY1
1	1	0	1	SY1a
1	1	1	0	SY2
1	1	1	1	SY2a

SPR:

gpr-type
areg-type
ls-ctl
pla-ctl
cb-ctl
loop-ctl
pcr
status

areg-names

3	2	1	0	
0	0	0	0	A0
0	0	0	1	A1
0	0	1	0	T
0	0	1	1	TR
0	1	0	0	PP0
0	1	0	1	Aout
0	1	1	0	PP1
0	1	1	1	Dout
1	0	0	0	SX1
1	0	0	1	SX1a
1	0	1	0	SX2
1	0	1	1	SX2a
1	1	0	0	SY1
1	1	0	1	SY1a
1	1	1	0	SY2
1	1	1	1	SY2a

FIG. 6I

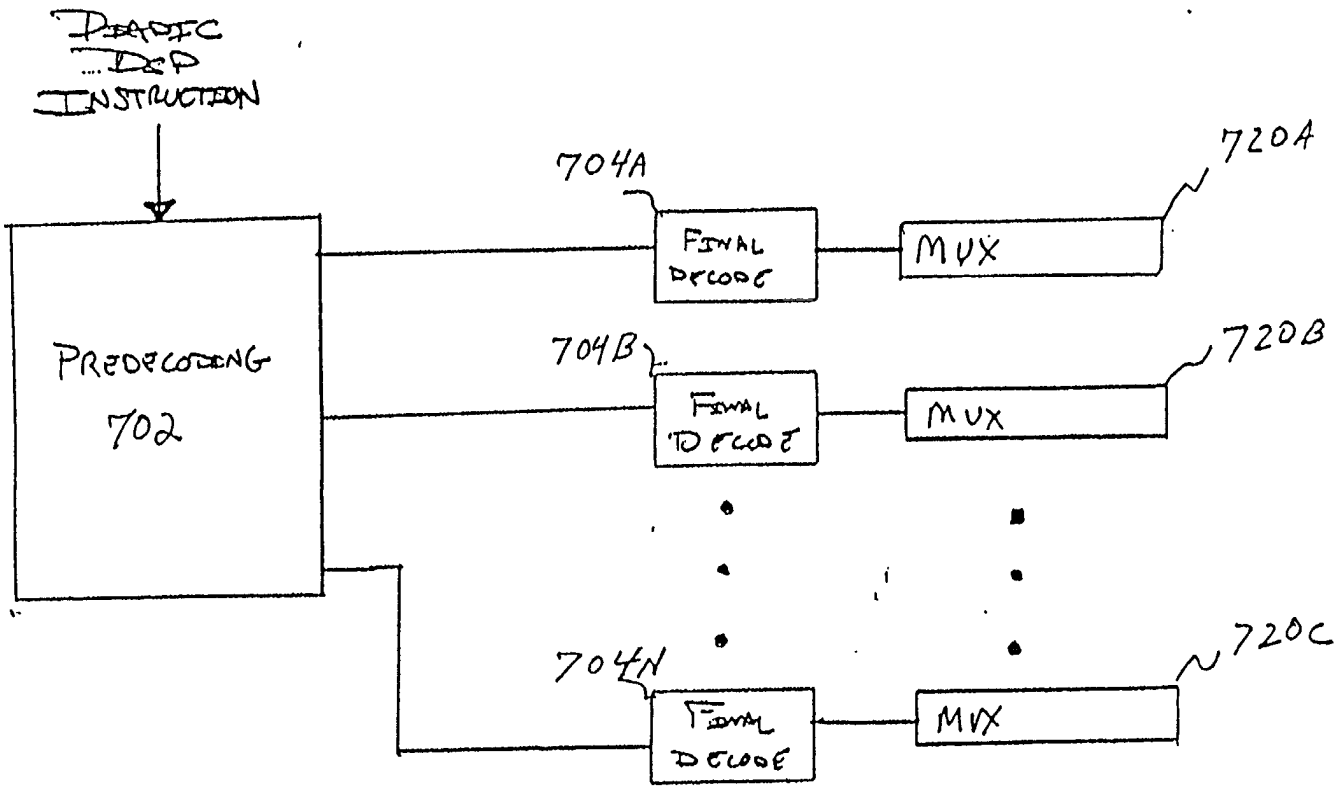


FIG. 7

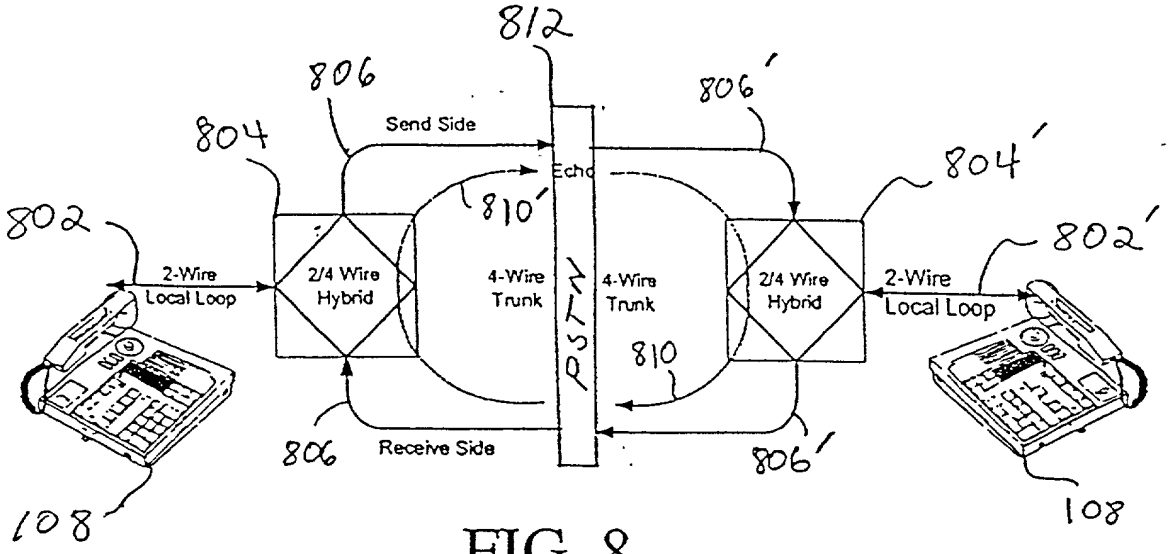


FIG. 8
(PRIOR ART)

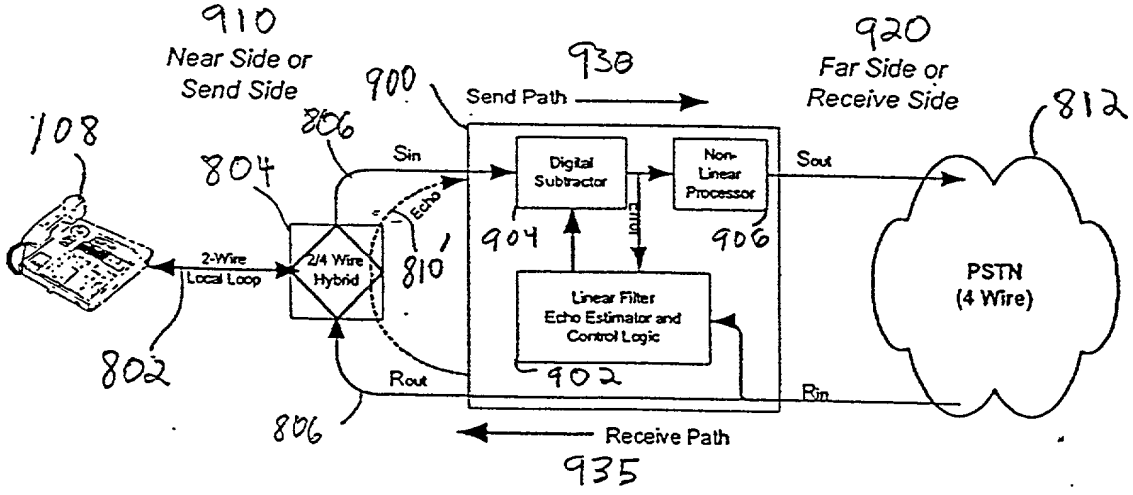


FIG. 9
(PRIOR ART)

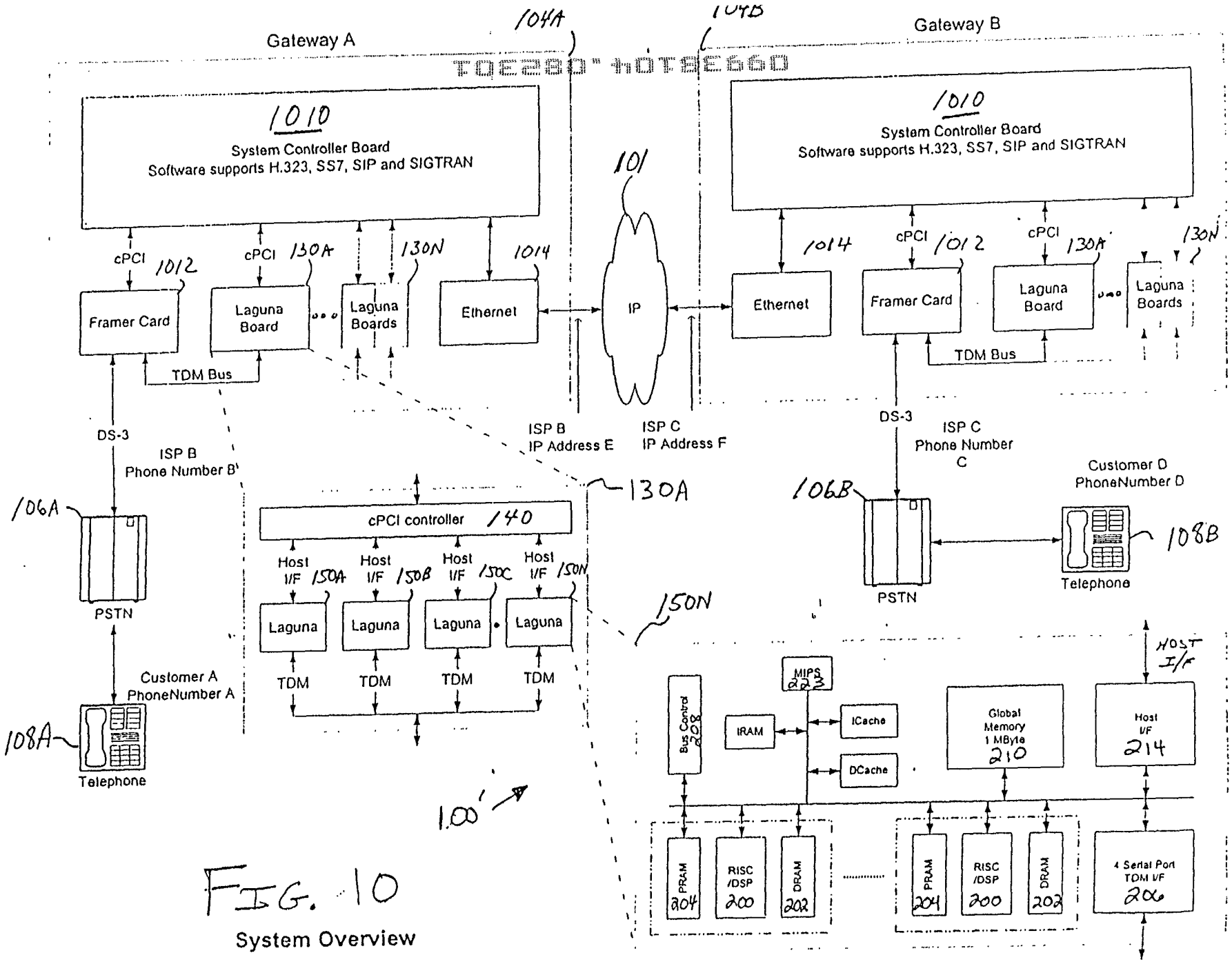


FIG. 10

System Overview

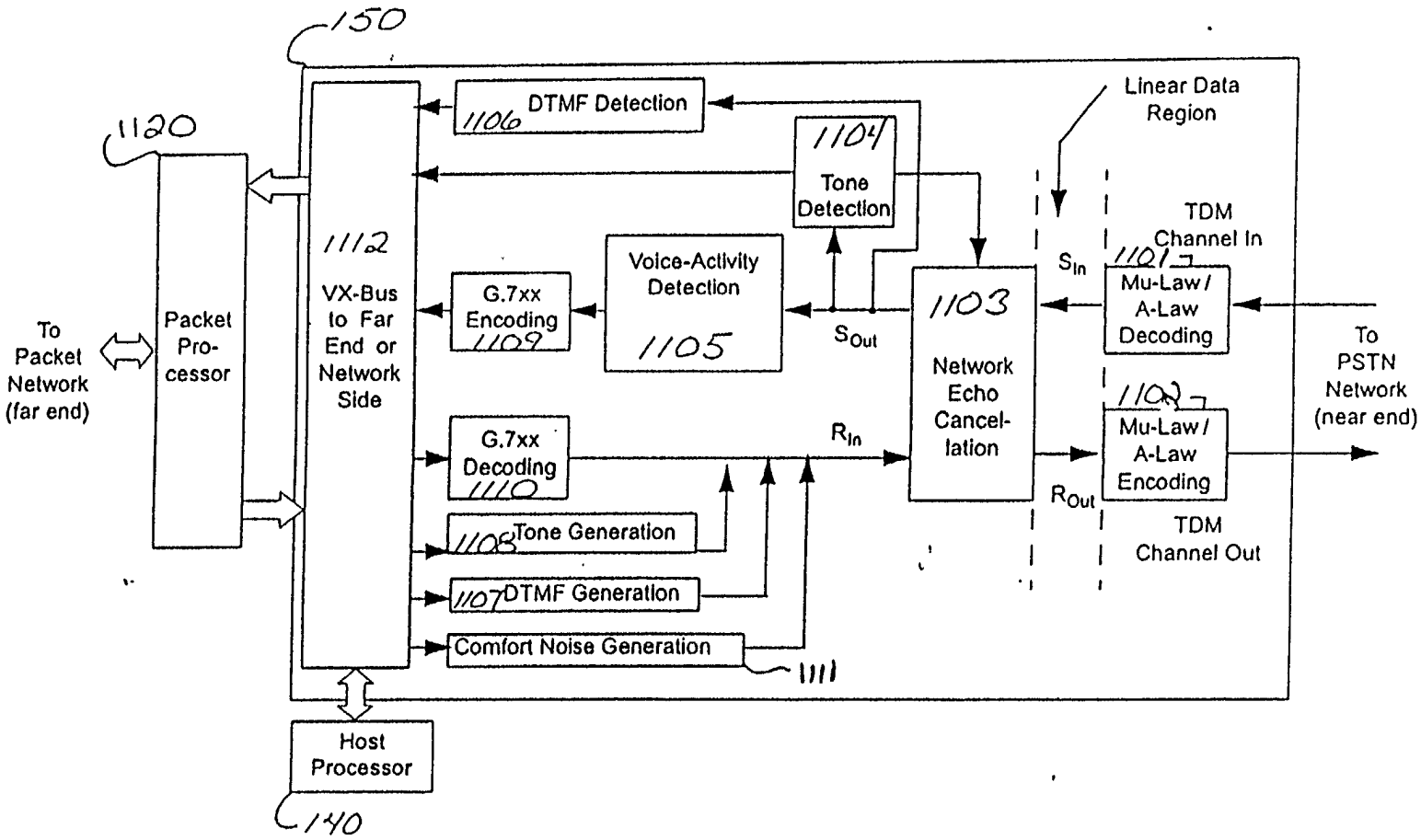


FIG. 11

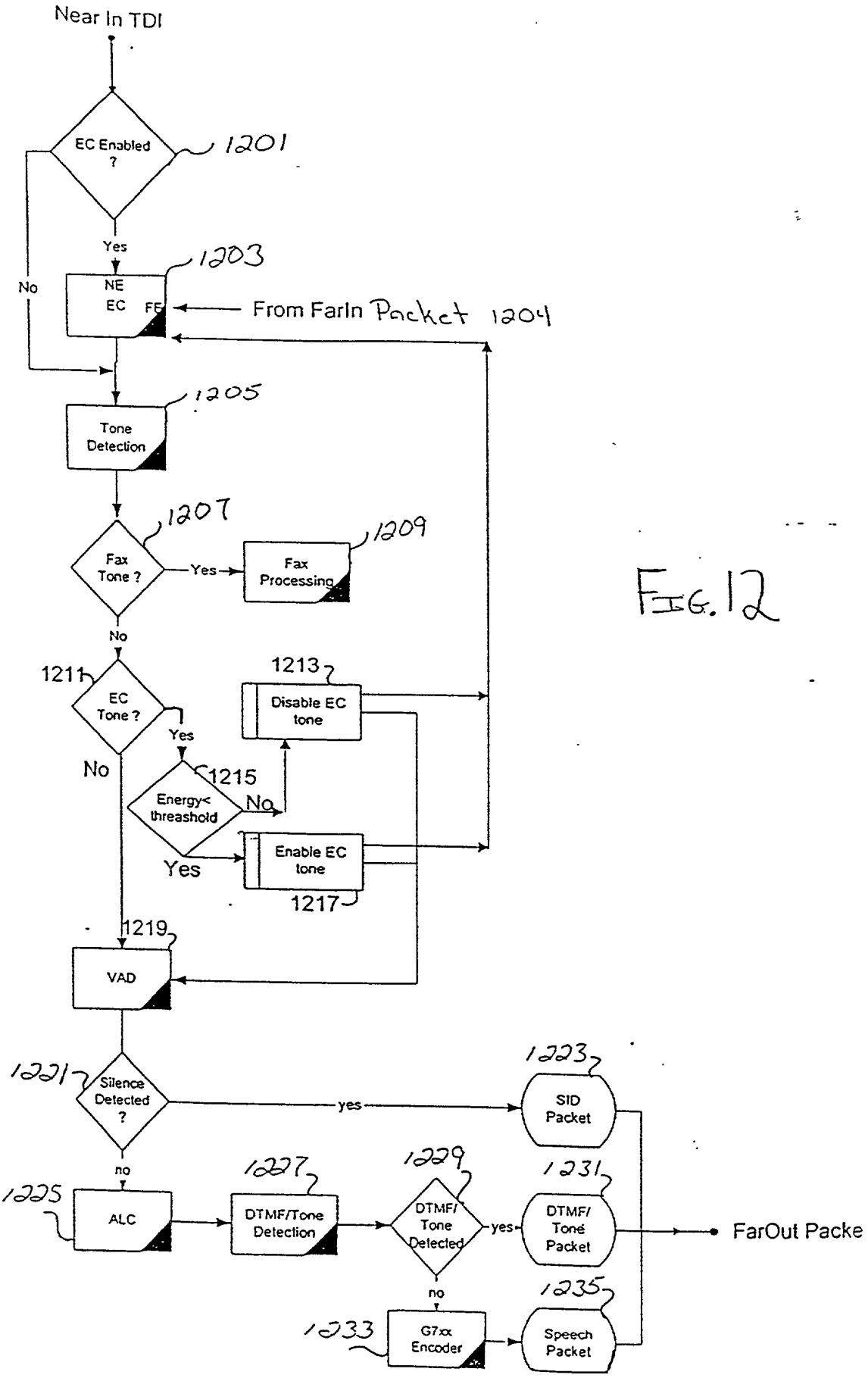

Near In TDI
EC Enabled ?
1201
Yes
No
1203
NE EC FE
From FarIn Packet 1204
1205
Tone Detection
1207
Fax Tone ?
Yes
1209
Fax Processing
No
1211
EC Tone ?
Yes
No
1213
Disable EC tone
1215
Energy< threashold
No
Yes
Enable EC tone
1217
1219
VAD
1221
Silence Detected ?
yes
1223
SID Packet
no
1225
ALC
1227
DTMF/Tone Detection
1229
DTMF/ Tone Detected
yes
1231
DTMF/ Tone Packet
FarOut Packe
no
1233
G7xx Encoder
1235
Speech Packet

FIG. 12

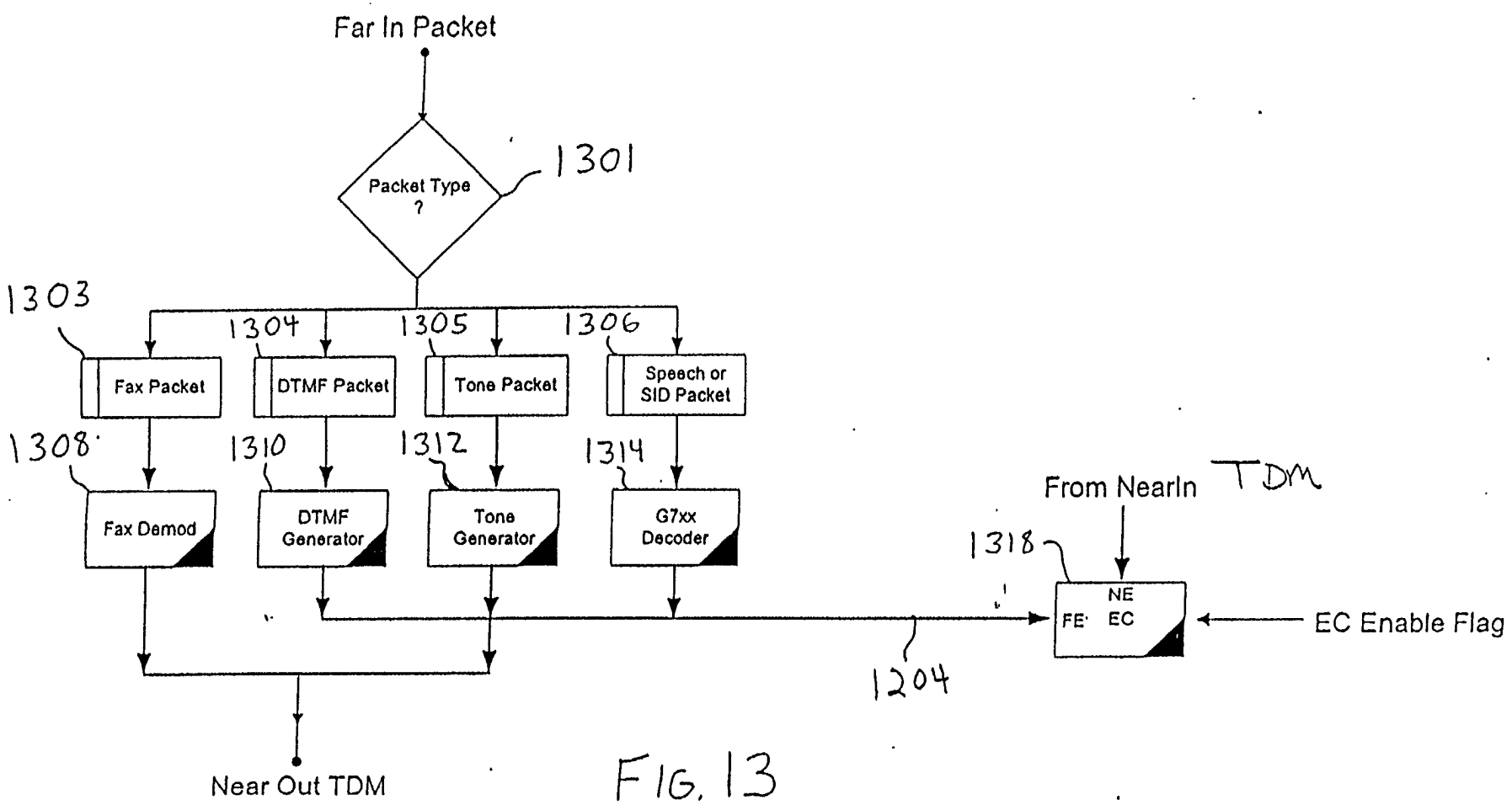


FIG. 13

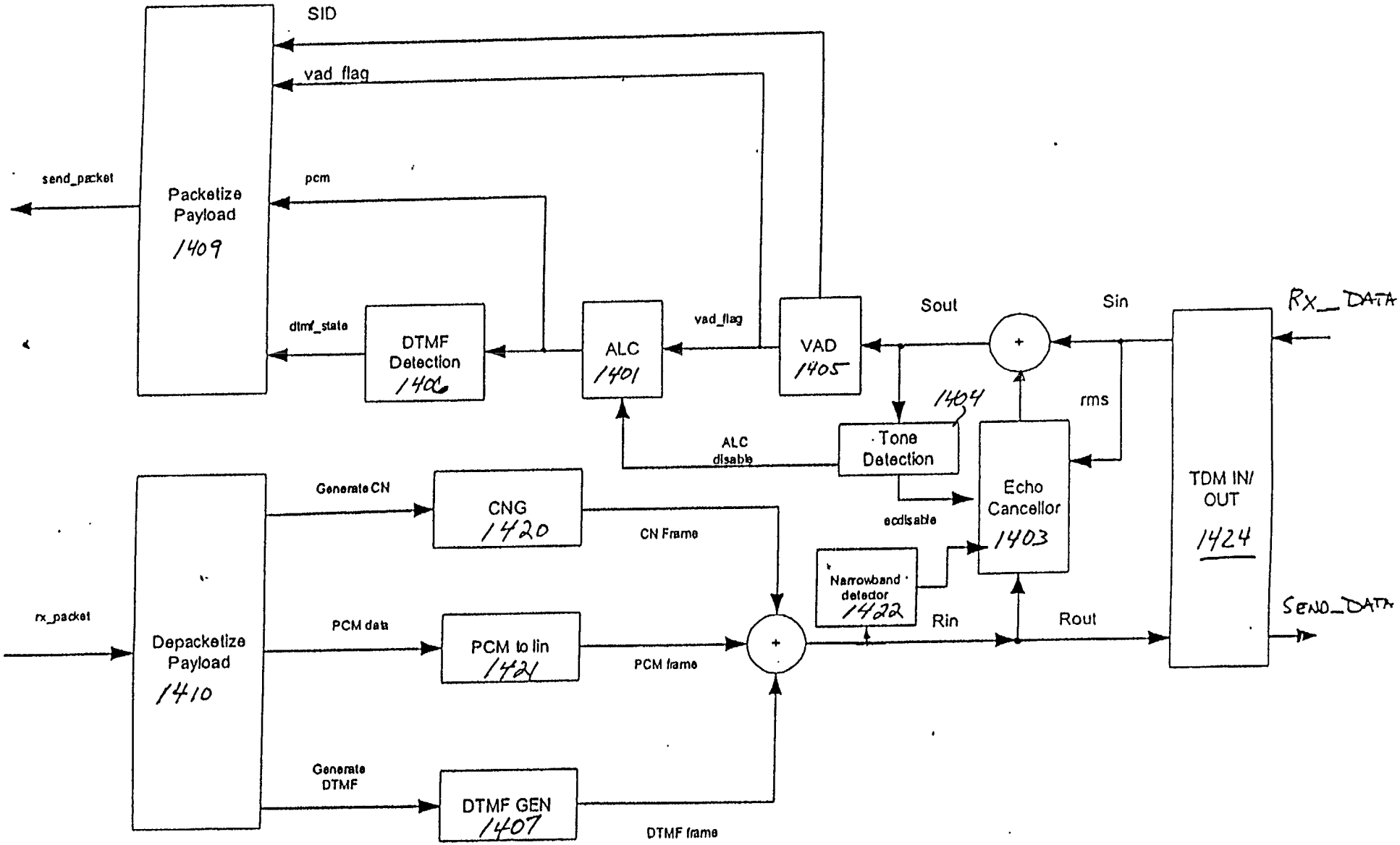

SID
vad_flag
send_packet
Packetize Payload
1409
pcm
dtmf_state
DTMF Detection
1406
ALC
1401
vad_flag
VAD
1405
Sout
Sin
RX_DATA
1404
rms
ALC disable
Tone Detection
Echo Cancellor
1403
TDM IN/ OUT
1424
Generate CN
CNG
1420
CN Frame
ecdisable
Narrowband detector
1422
rx_packet
Depacketize Payload
1410
PCM data
PCM to lin
1421
PCM frame
Rin
Rout
SEND_DATA
Generate DTMF
DTMF GEN
1407
DTMF frame

FIG. 14A

VxTel Voice Activity Detection Algorithm

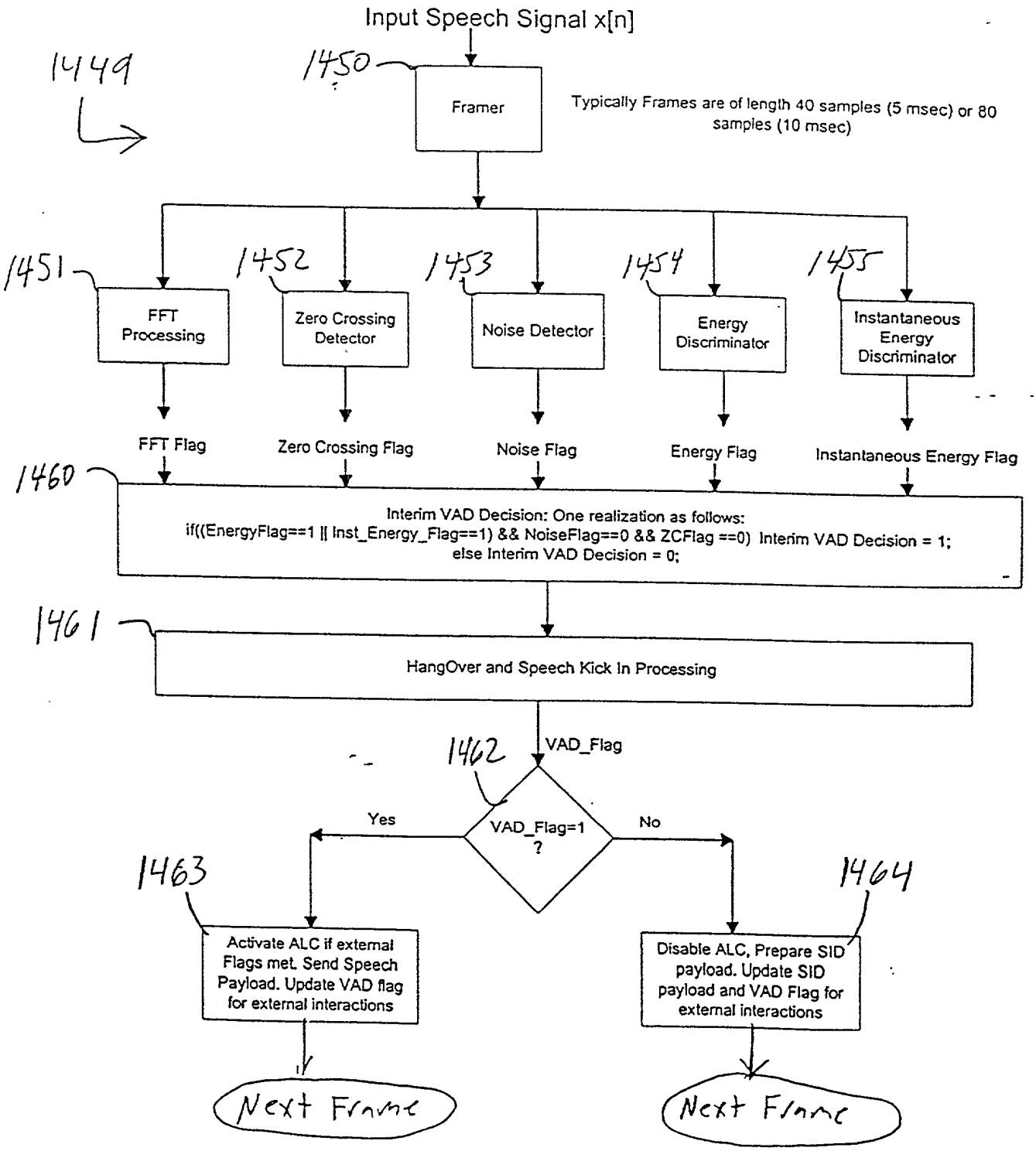


FIG. 14B

FFT Processing of Input Speech for VAD 1451

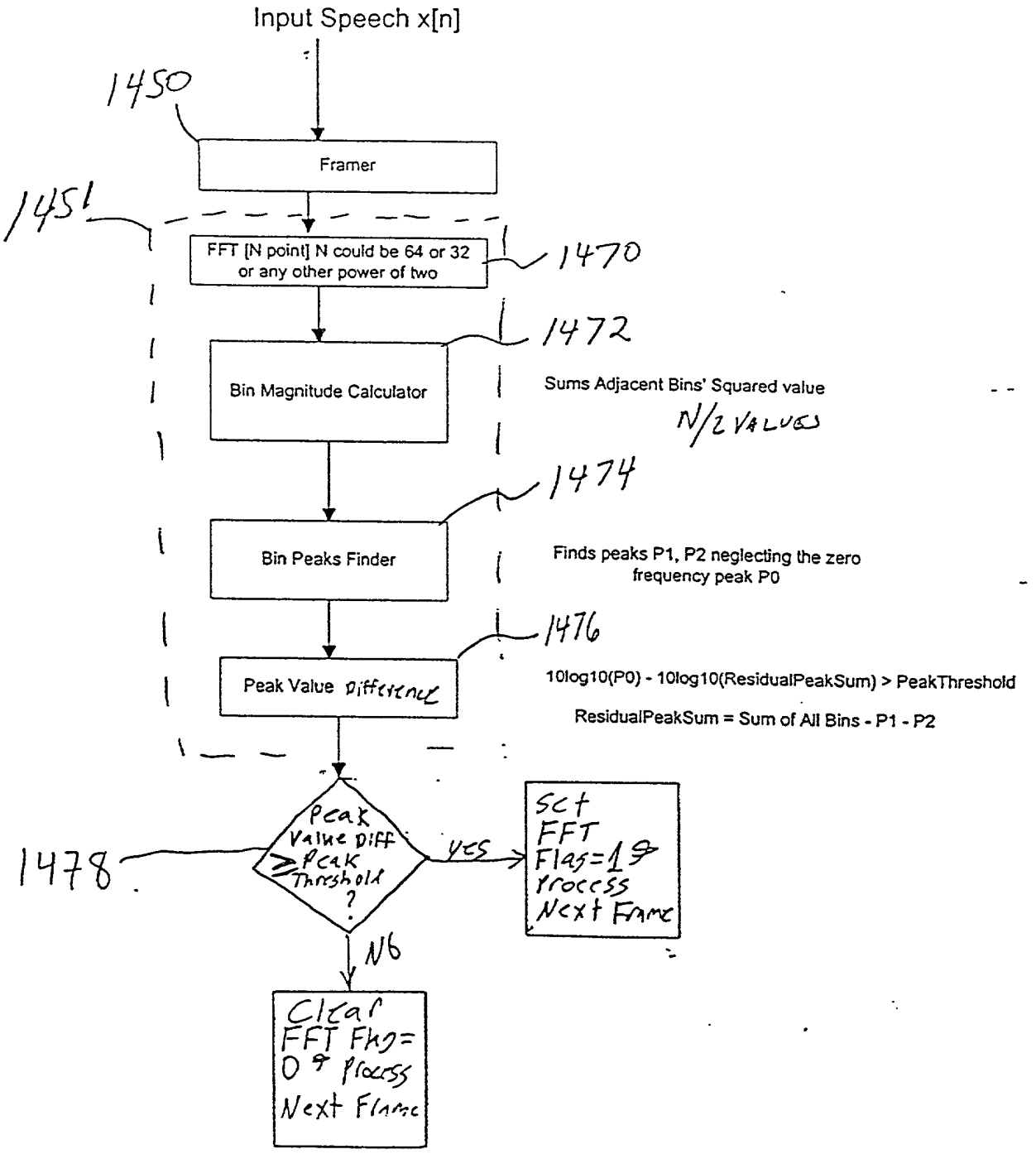


FIG. 14C

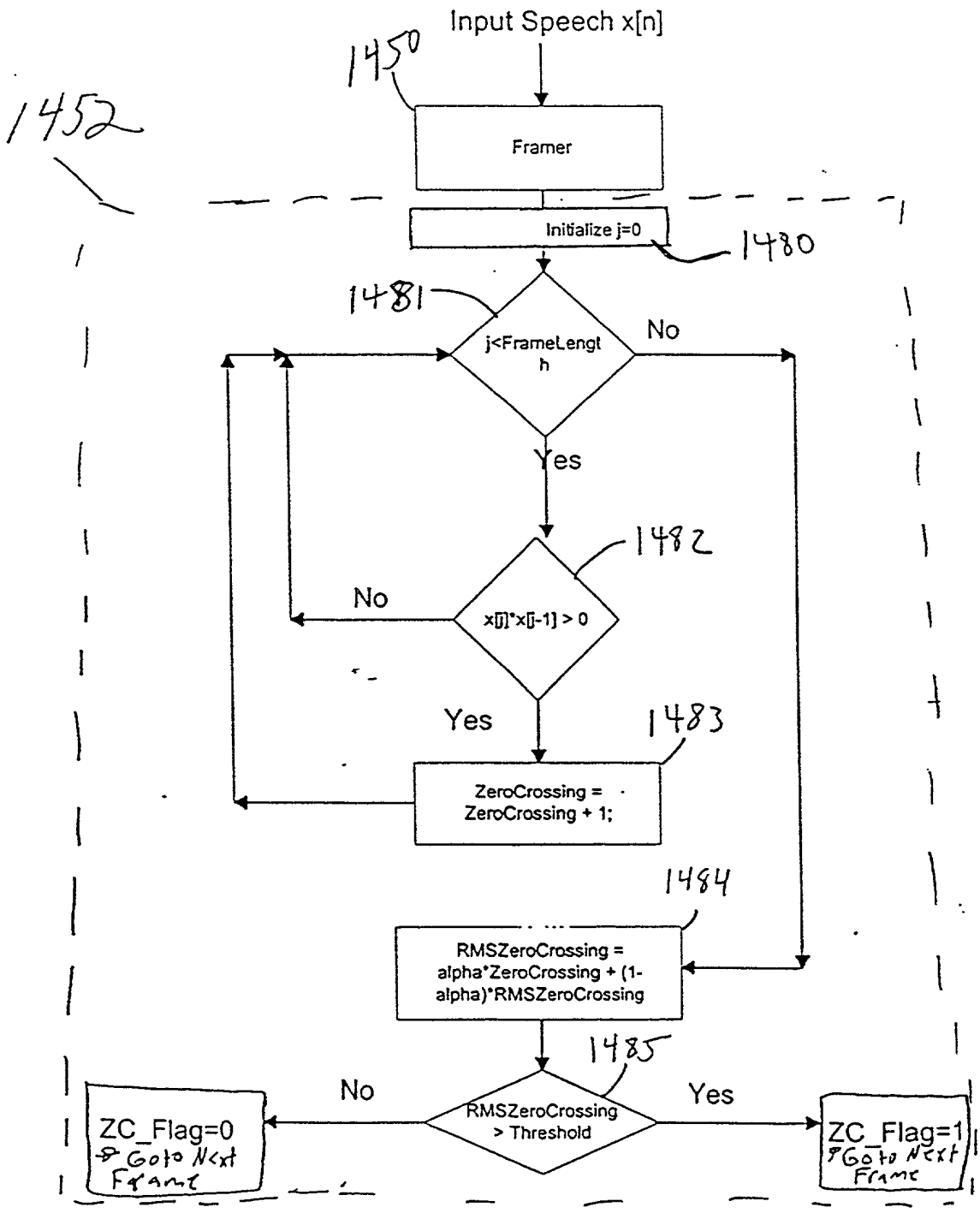


FIG. 14D

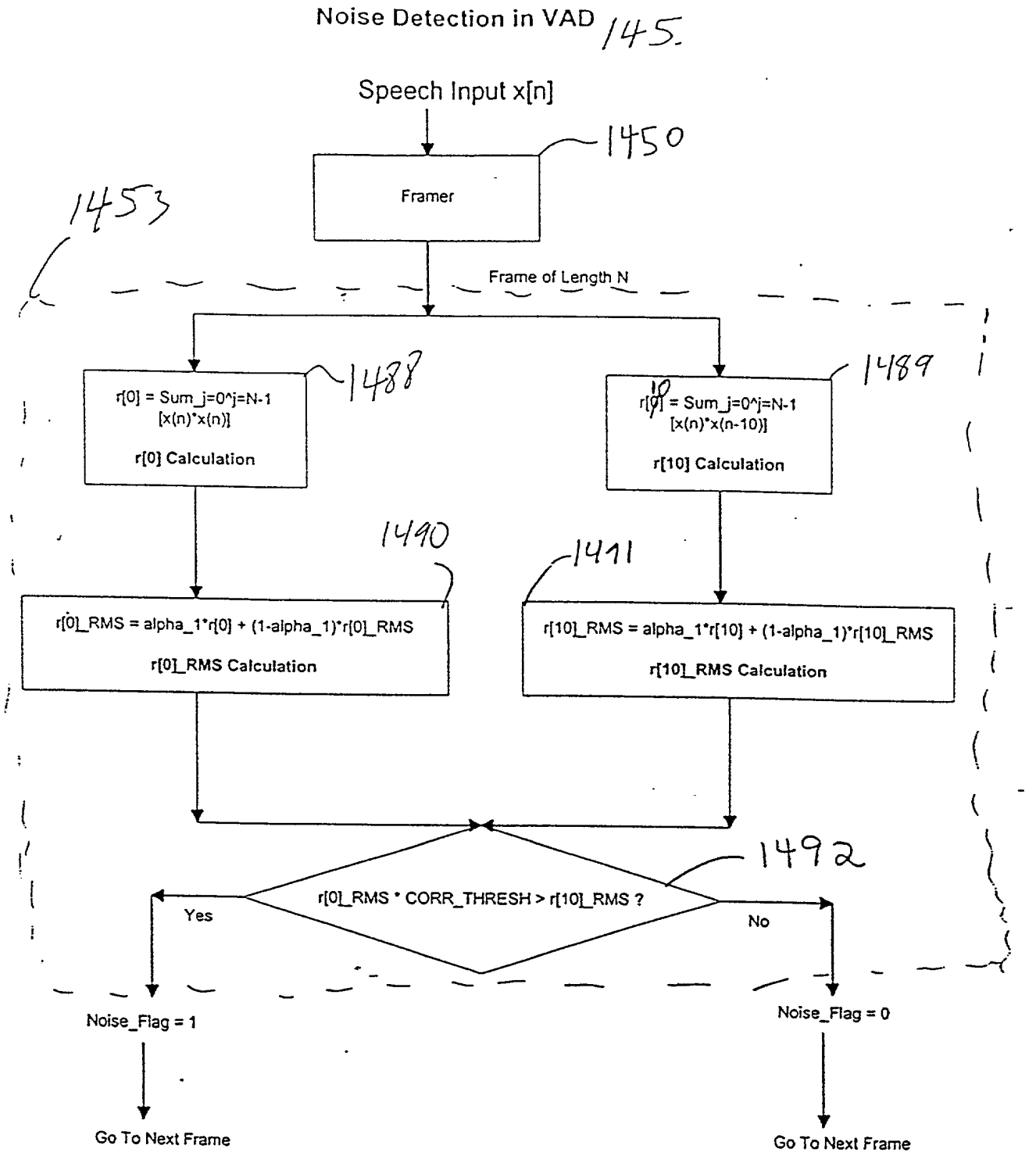

Noise Detection in VAD
145
Speech Input x[n]
1450
Framer
1453
Frame of Length N
1488
r[0] = Sum_j=0^j=N-1 [x(n)*x(n)]
r[0] Calculation
1489
r[10] = Sum_j=0^j=N-1 [x(n)*x(n-10)]
r[10] Calculation
1490
r[0]_RMS = alpha_1*r[0] + (1-alpha_1)*r[0]_RMS
r[0]_RMS Calculation
1411
r[10]_RMS = alpha_1*r[10] + (1-alpha_1)*r[10]_RMS
r[10]_RMS Calculation
1492
r[0]_RMS * CORR_THRESH > r[10]_RMS ?
Yes
No
Noise_Flag = 1
Noise_Flag = 0
Go To Next Frame
Go To Next Frame

Fig. 14E

Energy Discriminator 1454

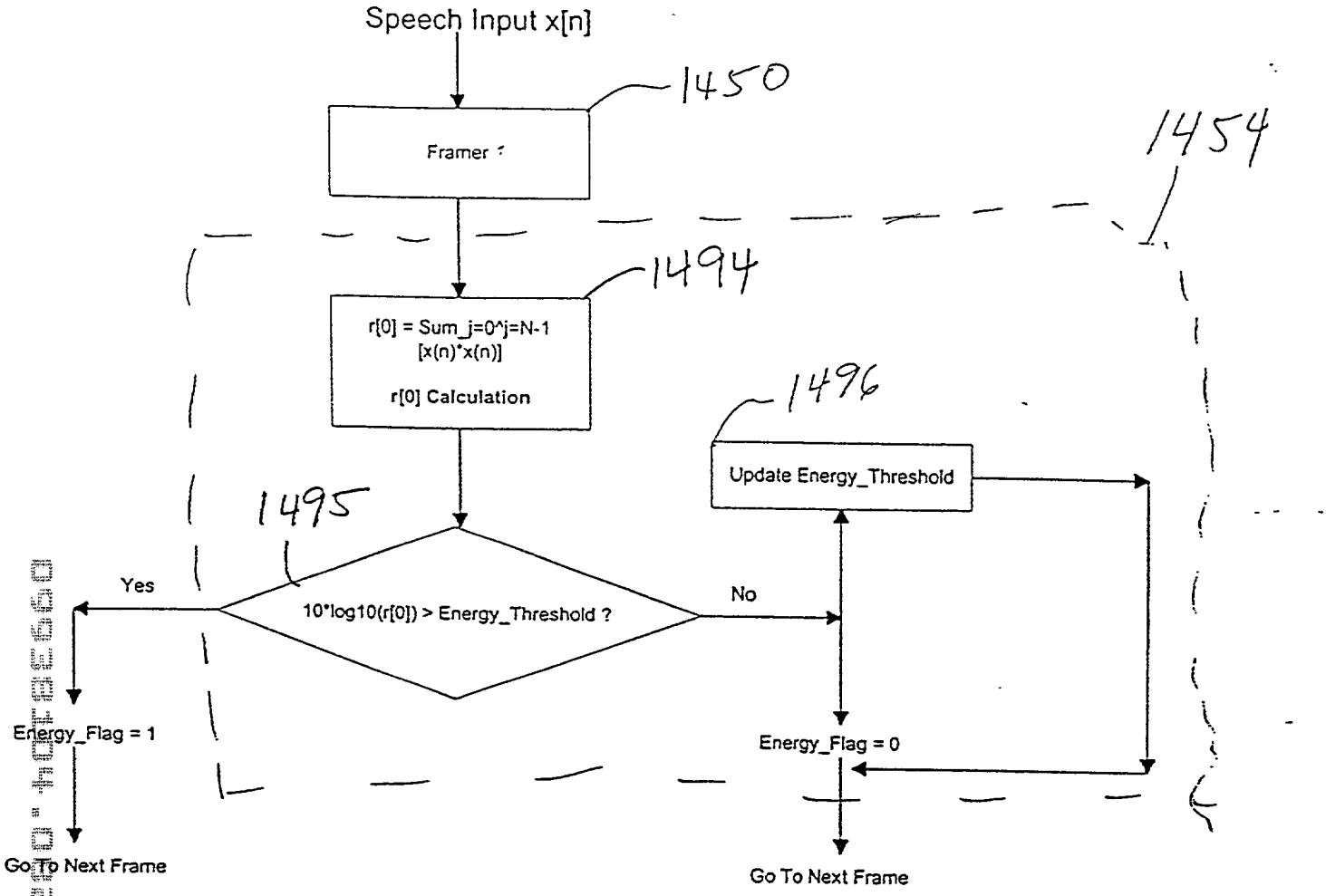


FIG. 14F

Instantaneous Energy Discriminator 1455

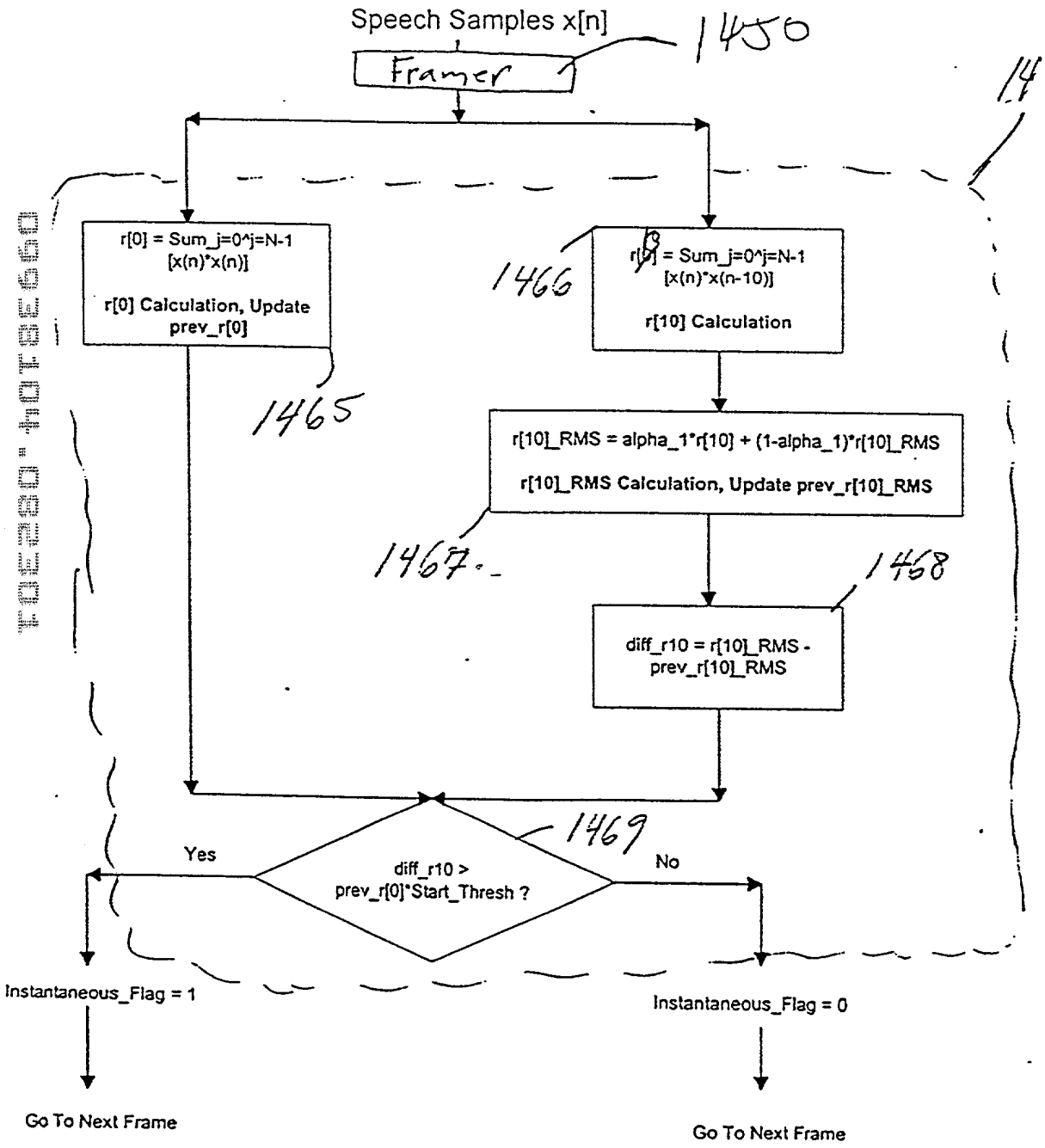


FIG. 14G

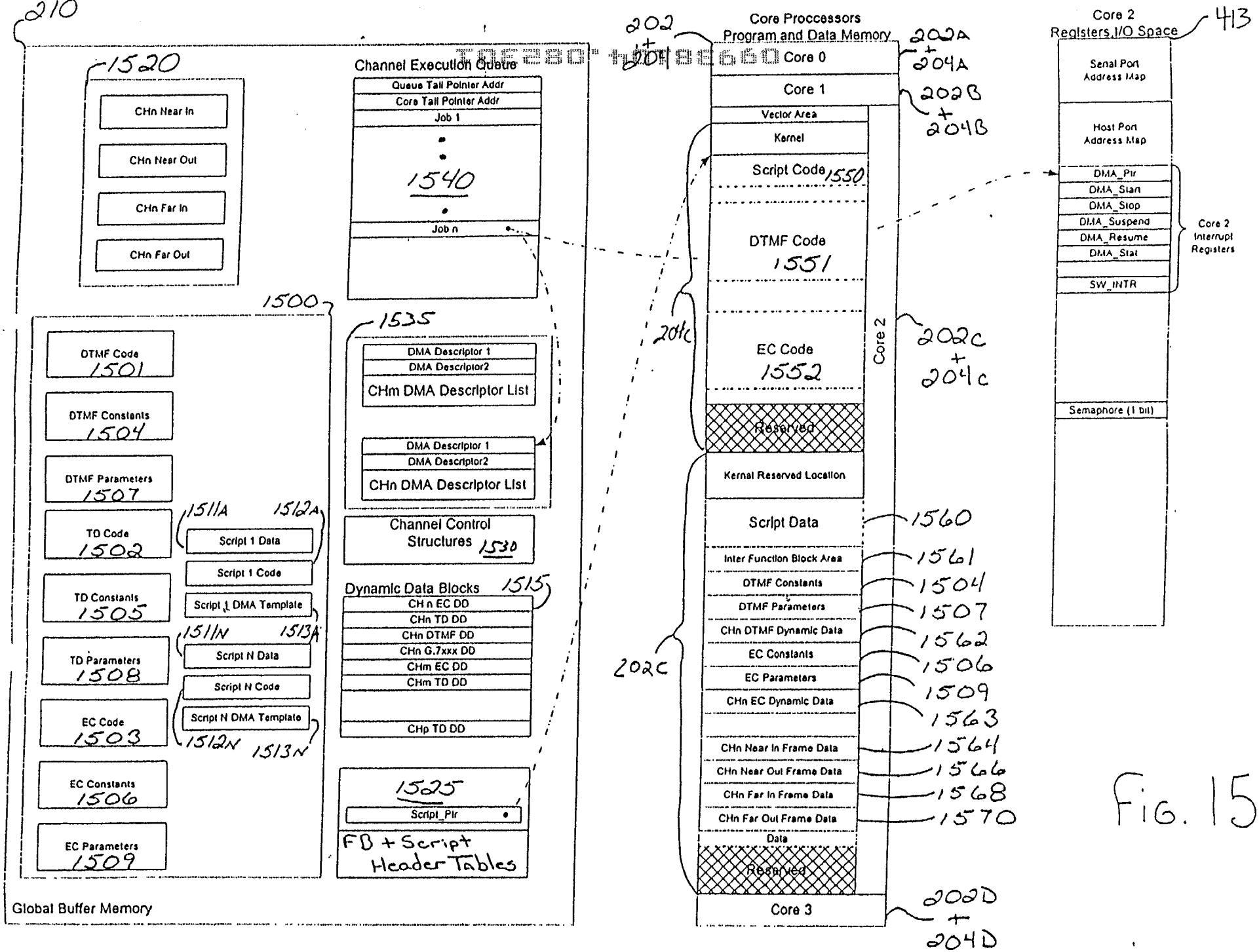

210
1520
CHn Near In
CHn Near Out
CHn Far In
CHn Far Out
Channel Execution Queue
Queue Tail Pointer Addr
Core Tail Pointer Addr
Job 1
1540
Job n
1500
DTMF Code 1501
DTMF Constants 1504
DTMF Parameters 1507
TD Code 1502
TD Constants 1505
TD Parameters 1508
EC Code 1503
EC Constants 1506
EC Parameters 1509
1511A
1512A
Script 1 Data
Script 1 Code
Script 1 DMA Template
1511N
1513A
Script N Data
Script N Code
Script N DMA Template
1512N
1513N
1535
DMA Descriptor 1
DMA Descriptor2
CHm DMA Descriptor List
DMA Descriptor 1
DMA Descriptor2
CHn DMA Descriptor List
Channel Control Structures 1530
Dynamic Data Blocks
1515
CH n EC DD
CHn TD DD
CHn DTMF DD
CHn G.7xxx DD
CHm EC DD
CHm TD DD
CHp TD DD
1525
Script_Ptr
FB + Script Header Tables
Global Buffer Memory
202
Core Proccessors Program and Data Memory
Core 0
Core 1
Vector Area
Kernel
Script Code 1550
DTMF Code 1551
EC Code 1552
Reserved
Kernal Reserved Location
Script Data
Inter Function Block Area
DTMF Constants
DTMF Parameters
CHn DTMF Dynamic Data
EC Constants
EC Parameters
CHn EC Dynamic Data
CHn Near In Frame Data
CHn Near Out Frame Data
CHn Far In Frame Data
CHn Far Out Frame Data
Data
Reserved
Core 3
Core 2
202A + 204A
202B + 204B
202C + 204C
202D + 204D
201c
202c
1560
1561
1504
1507
1562
1506
1509
1563
1564
1566
1568
1570
Core 2 Registers I/O Space
413
Serial Port Address Map
Host Port Address Map
DMA_Ptr
DMA_Start
DMA_Stop
DMA_Suspend
DMA_Resume
DMA_Stat
SW_INTR
Core 2 Interrupt Registers
Semaphore (1 bit)

Fig. 15

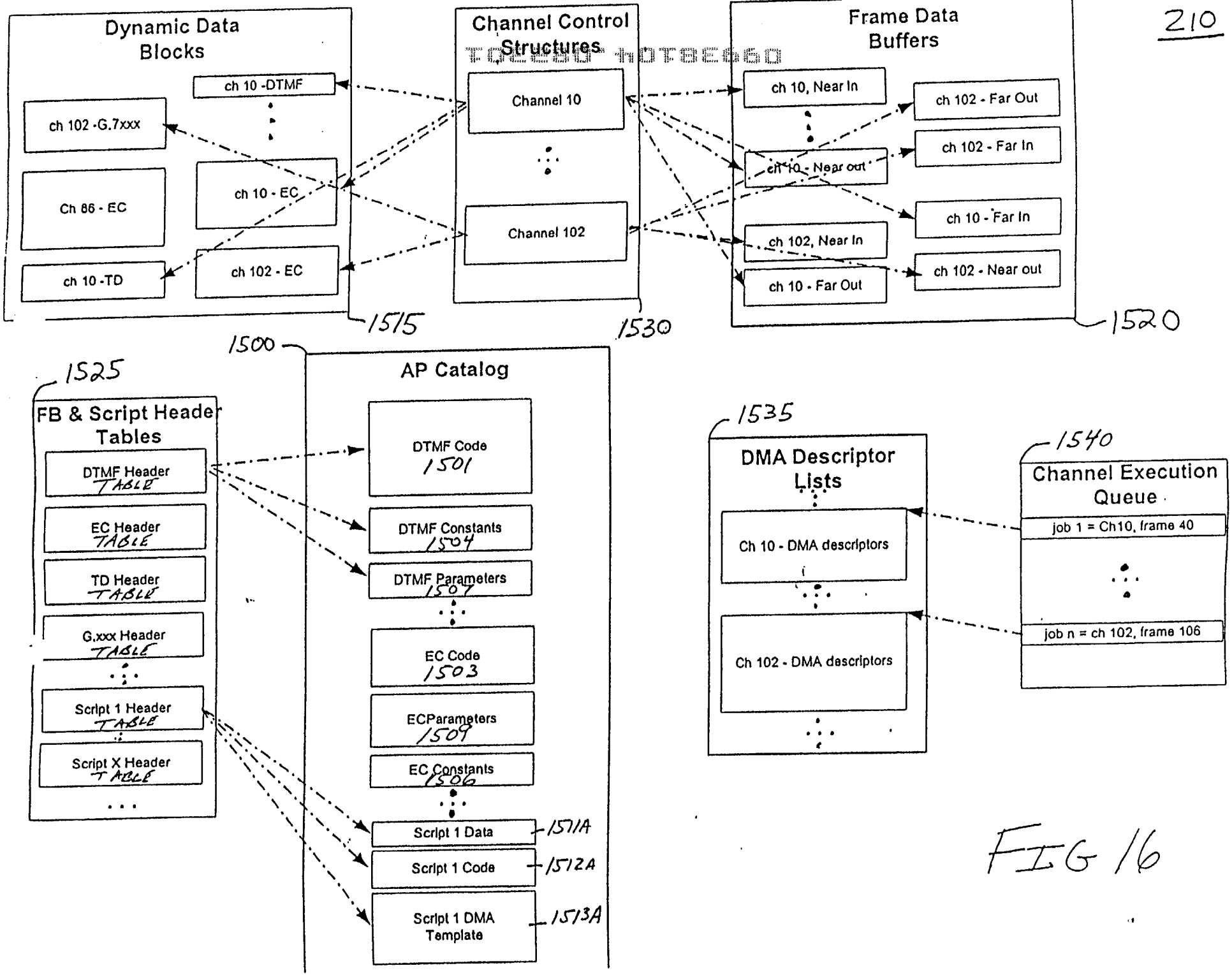

210
Dynamic Data Blocks
ch 10 -DTMF
ch 102 -G.7xxx
Ch 86 - EC
ch 10 - EC
ch 10 -TD
ch 102 - EC
1515
Channel Control Structures
Channel 10
Channel 102
1530
Frame Data Buffers
ch 10, Near In
ch 10 - Near out
ch 102, Near In
ch 10 - Far Out
ch 102 - Far Out
ch 102 - Far In
ch 10 - Far In
ch 102 - Near out
1520
1500
1525
FB & Script Header Tables
DTMF Header TABLE
EC Header TABLE
TD Header TABLE
G.xxx Header TABLE
Script 1 Header TABLE
Script X Header TABLE
AP Catalog
DTMF Code 1501
DTMF Constants 1504
DTMF Parameters 1507
EC Code 1503
ECParameters 1509
EC Constants 1506
Script 1 Data
1511A
Script 1 Code
1512A
Script 1 DMA Template
1513A
1535
DMA Descriptor Lists
Ch 10 - DMA descriptors
Ch 102 - DMA descriptors
1540
Channel Execution Queue
job 1 = Ch10, frame 40
job n = ch 102, frame 106

FIG 16

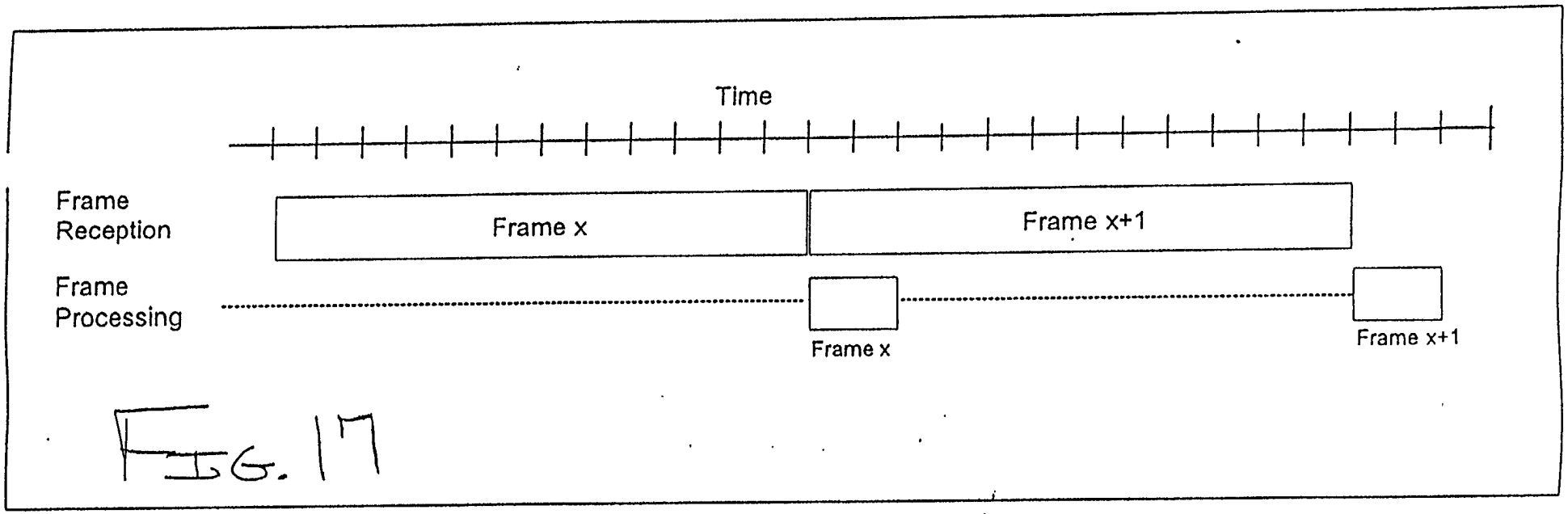


FIG. 17

FIG. 18